This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 636427.

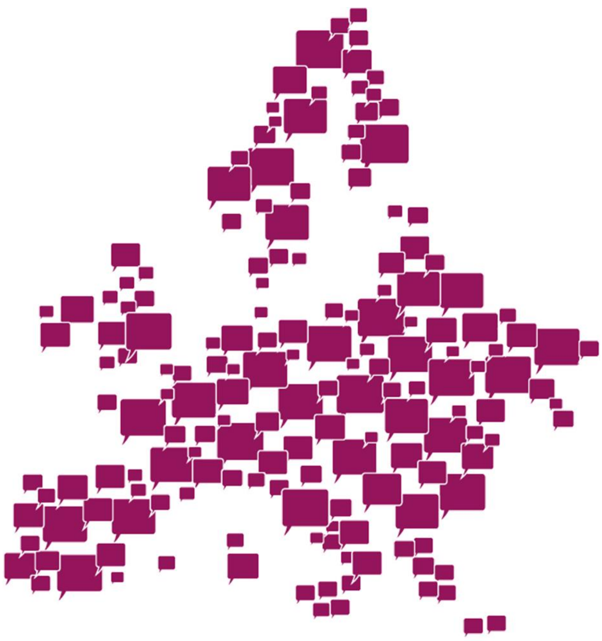
socialcar

www.socialcar-project.eu

SocialCar

Open social transport network for urban approach to carpooling

Social transport graph – Route planning and ride matching - 1



Deliverable D2.1	
Dissemination level: public	
Work Package: 2	
Version: 0.1	
Date of preparation: 15.01.2016	

Table of contents

1 Introduction

1.1 Scene setter

This deliverable aims to lay the foundations for the development of the framework and the algorithms which will power the SocialCar platform. On the basis of this work, WP2 will proceed to a first implementation of the algorithms, in order to set up an initial working prototype of the whole SocialCar Platform. This deliverable will be updated in month 30, as described in the project work plan, in order to refine and complete the initial concepts outlined in this document. The output of this deliverable is aimed at both WP3 and WP4.

Project	SocialCar: Open social transport network for urban approach to carpooling
Grant agreement number	636427
Document title	Social transport graph – Route planning and ride matching - 1
Deliverable number	D 2.1
Type of deliverable	Report
Dissemination level	public
Work Package	WP2
Version	01
Date of preparation	15 January 2016
Language	English
Author(s) (Beneficiary)	Andrea Emilio Rizzoli, Roberto Montemanni (SUPSI), Circe Serra (EURECAT), Flavio Minori (MOVENDA), Dionisis Kehagias (CERTH), Emma Silk (LIFTSHARE), Maria Bulgheroni, Chiara Bonizzi (Ab.ACUS)
Contributor(s) (Beneficiary)	Angelo Meuleman, Luk Roose (TAXISTOP), Kornel Czimber (UWH)
Peer reviewer(s) (Beneficiary)	Fabio Cartolano (FIT), Dionisis Kehagias (CERTH)
History	V 0.1 description of the deliverable content V 0.2 description of data requirements and initial review of state of the art in the algorithms
Keywords	Data model, rail road network model,

1.2 Executive summary

This deliverable contains the data model for the SocialCar platform. It represents a multilayer temporal network for both public and private transport. It integrates the representation of car pooling services. The proposed data model describes users in movement offering transport solutions, and users requesting transport services. It also includes information about their reliability and trust in the use and provision of car pooling services.

The deliverable also contains a description of the algorithms to be used in the SocialCar platform. It describes the dynamic route planning, ride matching, tracking and destination tagging algorithms.

It defines the algorithms to measure and update user reputation and the reward and incentive mechanisms.

This document is structured as follows:

- Chapter 1 is this chapter;
- Chapter 2 contains a description of the data structures that are required to develop the SocialCar platform. In particular it deals with static data, such as road and rail infrastructures, and seasonal public transport schedules, which are changed only once or twice per yar, and dynamic data, such as the state of transport networks, changes and deviations in arrival and departure times of public transport, and finally car pooling data, in terms of transport demand, and transport supply.
- Chapter 3 describes, from a broad view point, the various algorithms that can be implemented in the SocialCar platform. It does not contain the final choices, but it presents a review of the state of the artthat has been done from the SocialCar perspective. In particular, the algorithms for route planning and ride matching, for user evaluation and reputation, and for user tracking, are described.
- Chapter 4 makes a summary of the deliverable presenting the conclusions.

1.3 Scope of the document

Provide an initial description of the data models and the algorithms to be used in the SocialCar backend and frontend applications.

1.4 Glossary

CSV	Comma Separated Value, a format for text files.
GTFS	General Transit Feed Specification, a standard specification for public transport routes and timetables.
OSM	Open Street Map, open source and collaborative road network data and attributes.
OTP	OpenTripPlanner, a software platform for multimodal trip planning, based on OSM and GTFS.
POI	Point(s) of interest.
TMC	Traffic Message Channel, a technology to deliver traffic and travel information to drivers.
TMS	Traffic Management System, the system composed by sensor data, analytical instruments, simulation models, used by a city to manage urban traffic.

2 Data overview

This section describes the data structures to be made available as input to the SocialCar algorithms, in order to offer the SocialCar services.

SocialCar operates with different types of data, which we have logically organised in two main categories, the Supply side data and the Demand side data.

The supply side includes:

- Transport infrastructures:
    - Road networks, including bike paths, pedestrian paths.
    - Rail networks, including light rail.
- Transport service data:
    - Static: bus lines, train schedules.
    - Dynamic: car pooling services, taxi[1].
    - On demand: taxi, bike sharing, car sharing.
- Transport status:
    - Traffic intensity, level of service.
    - Rail network status/delays.

The demand side includes:

- Transport requests from origin to destination, its status.
- User related data (position, reputation, etc.).

SocialCar aims to solve a multimodal route planning problem, i.e., a route planning problem involving different transportation modes. To solve such a problem, high quality data related to the different transport options must be collected and integrated.

A description of the data available in the SocialCar test sites is presented in Deliverable D2.3. In general, where no other data source is available, the transport infrastructure data (road and rail networks) can be obtained from OpenStreetMap[2], while the static transport service data (bus and train schedules) is available as GTFS data (General Transit Feed Specification)[3] for many locations around the world. For instance, OpenTripPlanner[4], an open source project, uses standard GTFS data. Real time transport information can be also made available in the GTFS-realtime format.

In order to integrate all the various multimodal transport possibilities, the logical organisation of data described above needs to be structured as a multi-layer temporal network, as described in Gallotti and Barthelemy (2015). The multimodal network model is strictly related to the algorithms that can be used to solve the planning problem, so its design and implementation depends on the specific algorithms to be applied to it.

[1] Taxi services have a low priority in SocialCar

[2] http://www.openstreetmap.org

[3] http://www.gtfs-data-exchange.com/agencies/bylocation

[4] http://www.opentripplanner.org

In the remainder of this section we introduce the main concepts of the network model that will be used by the SocialCar algorithm for route planning.

2.1 Multi layer temporal networks for SocialCar

The SocialCar algorithm needs an underlying network in order to create alternative route solutions for users willing to move from origin to destination selecting alternative transport means.

The network must be made available as a **graph**, composed by **nodes**, which are connected by **edges**. The nodes represent the junctions in the network, from where links depart. A link connects two nodes.

The network is composed by **layers**. Each layer represents a different transport mode. Nodes that are present in multiple layers simultaneously represent intermodal connection points (e.g. bus and train stops). There are also interlayer links which represent the travelling time (by foot) required to transit from a transport mode to another. The travelling time does not include the time eventually required to purchase a ticket and it assumes a slow walking speed, as stairs and escalators might be required. The nodes where a modal change can take place have been defined as **switch points** in the work of Liu (2011).

The layers associated with scheduled transport are defined as **temporal networks** (see Holme and Sarämaki, 2012): each edge in the graph modelling the network represents a segment in a route, while the nodes are the stops/stations.

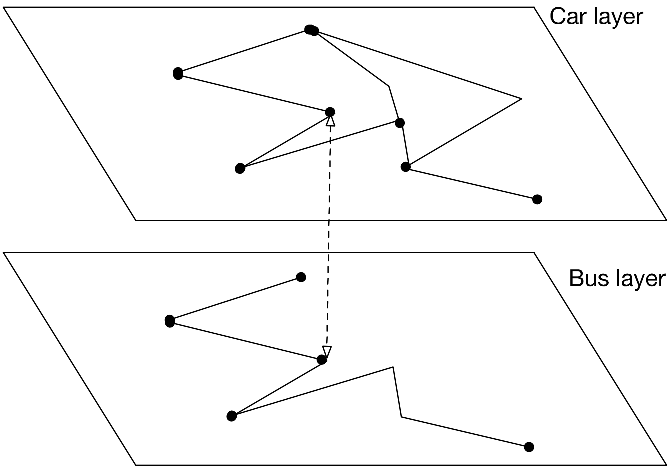


Figure 1. Two different layers contain the graphs for car and bus routes. The dashed lines represent an interlayer link, which allows a modal change across a selected route.

The considered transport modes are:

- **Foot**, it is the network of all roads that can be walked.
  Each link in the foot network has the following attributes:
    - Length (in metres).
    - Duration (according to an average walking speed of 4 km/h).
- **Bike**, it is the network of all roads that can be walked, and some additional roads that are dedicated only to bike. Roads that are only for pedestrians are marked so, but the bike can be pushed on those stretches. A common limitation of routing algorithms for bikes is that they do not foresee the possibility that the

rider dismounts the bikes and carries it around, even on stairs.
Each link in the bike network has the following attributes:

- Length (in metres)
- Duration (according to an average cycling speed of 15 km/h)
- Pedestrian: Boolean attribute indicating that the bike must be pushed by hand. All one-ways are labelled as "pedestrian" if ridden in the forbidden direction.

- **Car** (and motorbike), it is the network of roads that can be driven by car (motorbikes are assimilated to cars). The network is represented as a directed graph, where the nodes represent junctions and edges connect nodes. Each edge in the car network has the following attributes:
  - Length (in metres).
  - Duration (according to an average speed calculated on the basis of the speed limit).
  - Direction (from-to node) if it is a one way.
- **Bus**, as said, this is a temporal network, because it is the network of roads that are driven by buses, but the links are active only during specific time durations, which correspond to the bus timetable. The nodes in this network are only the bus stops. Each node is annotated with the departure times of each service departing from the node. For each service, there exists a list containing the sequence of nodes to be visited along the service route.
- **Train/Tram/Subway**, as above. It is a temporal network.

2.2 Modelling public transport as temporal networks

Modelling public transport networks requires an additional level of complexity with respect to the road networks, as routes are possible only when a service is scheduled. Moreover, in urban areas, the services are offered over transport lines, which partially overlap, and offer the traveller multiple chances to switch from a line to another, from a transport mode to another (e.g. from bus, metro, rail, tram to any other of the above).

In order to represent the transport services offered on top of the transport network, each node on the public transportation network is annotated with a list of departures times containing all the routes departing from that node.

For instance, see Table 1. The data refers to the Mendrisio railway station in Canton Ticino (CH), which is an intermodal hub, as it also serves as a Bus station. The table is organised by Transport ID ("Travel with" in the table), the departure time (Dep.), and the list of stations along the route (To). The data in the table is produced by a query returning the routes departing from a given station at a given time.

Table 1. An excerpt from the SBB timetable, based on the Hafas data.

Travel with	Dep.	To
NFB 3 222	12:09	Morbio Inferiore, Posta Mendrisio, Stazione- Mendrisio, Asilo Sud- Mendrisio, Centro Studi 1- Coldrerio, Colle degli Ulivi• Balerna, Piazza- Morbio Inferiore, Serfontana- Morbio Inferiore, Polenta- Morbio Inferiore, Ghitello- Morbio Inferiore, Sta. Lucia- Morbio Inferiore, Posta
NFB 1 14	12:11	Mendrisio, Cantine di Sotto Mendrisio, Stazione- Mendrisio, Macello- Mendrisio, Autosilo- Mendrisio, S. Giovanni- Mendrisio, Cantine Delizie- Mendrisio, Morgana- Mendrisio, Cantine di Sotto
BUS 531 17	12:12	Capolago, Stazione Mendrisio, Stazione- Mendrisio, Macello- Mendrisio, Via Morée- Mendrisio, S. Martino• Mendrisio, Vignalunga- Mendrisio, Vignöö- Mendrisio, Tre Antenne- Riva S. Vitale, Scuola Media- Capolago, Posta- Capolago, Stazione

2.2.1 Time-expanded graphs

The information of the table above is used to generate the temporal network representation: each ride from a stop/station to the next one is represented with a directed edge with duration equal to the time of arrival at the destination node minus the departure time at the origin. Each edge will be denoted by the name of the service. This is described in Figure 2 and Figure 3.

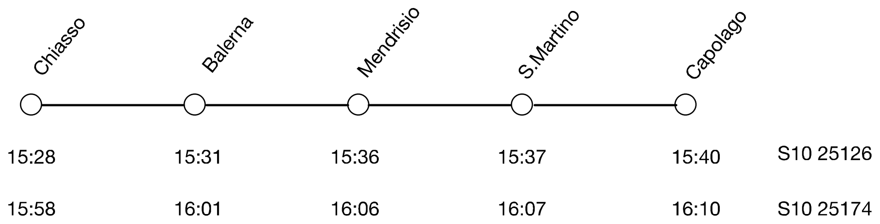


Figure 2. Two trains (S10 25126 and S10 25174) extracted from the timetable connecting the railway stations of Chiasso and Mendrisio.

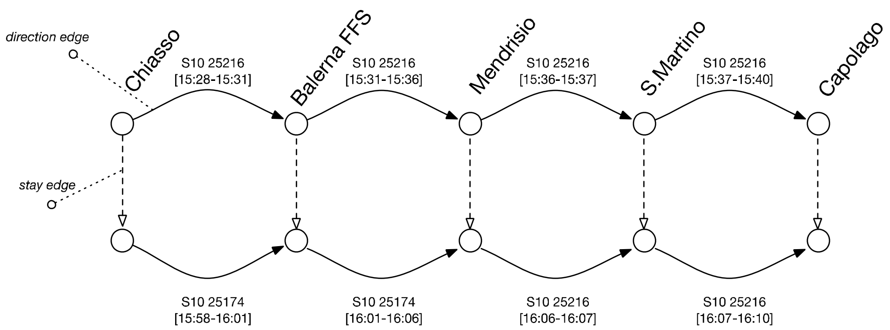


Figure 3. The timetable of Figure 2 is converted into a time expanded directed graph. Each edge has the departrure and arrival time between the consecutive nodes.

The above situation is the simplest case, even if there is a potential proliferation of arcs between nodes, especially on underground lines, which are travelled with a high frequency. Imagine a service running with a ten-minute frequency. In that case, we would potentially generate 6 services per hour, over possibly 20 hours of service, thus creating 120 arcs between each couple of nodes over the line.

Things can become more complex when we have stations with multiple lines where the user can change either transport line (e.g. from metro line A to metro line B) or transport mode (e.g. from metro line A to tramway line 121).

The main advantage of the above representation is that standard algorithms for the shortest path work with little or no modification, as the resulting network is static. A drawback is the large size of the resulting network (even if sparse), which can lead to longer computation times with respect to the time dependent case.

2.2.2 Time dependent graphs

Another approach to modelling public transit networks is based on time dependent graphs. This representation does not require proliferation of graph nodes in order to represent different departure times from a station. On the other hand, it uses the information contained in the timetables to generate the arc weights "on the fly" (Pyrga et al. 2008).

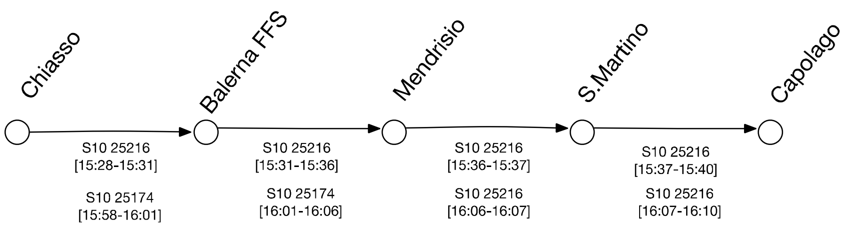


Figure 4. The timetable is used to dynamically calculate weights (travel times) on the graph.

Note that the time dependent model requires that a train departing from a given station after another train cannot overtake the preceding train. This constrain would prevent planning where InterCity or high speed trains are involved, but given that we are discussing urban mobility, we can say that this assumption is verified.

2.2.3 Switch nodes

In a multi layer network, each transport mode is represented by a different layer, and each transport line can also be represented as a different layer. Stations where the users can switch modes and lines have been defined as switch points in Liu (2011). Such a situation is described in Figure 5.

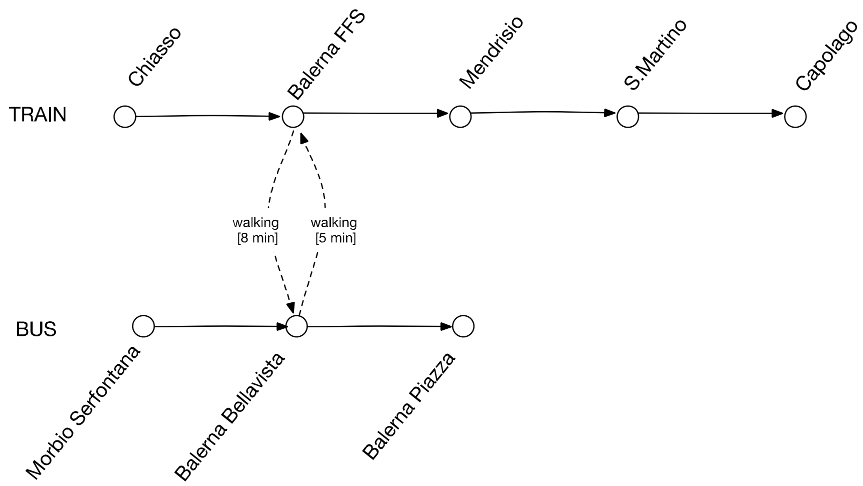


Figure 5.In Balerna the passenger can switch to the bus line. Balerna is therefore a switch node.

The road network is therefore considered as another layer on top of the previous ones, but no time constraints are fixed on the graph edges.

It has to be noted that a switch point might depend on the role of the user: a car driver must park the vehicle in an authorised parking place (e.g. Park and Ride), while a passenger can be dropped in front of a bus/train station even where long term car parks are not available. The situation is different for bike riders: it has to be specified by the bike owner if she accepts to park the bike in an unguarded bike stall, or if a warded bike park is a requisite: this selection can activate and deactivate different switch points.

2.2.4 Source data formats for representing public transport services

For the description of public transport services, the most obvious solution is to adopt the GTFS format. In brief, a GTFS feed is composed of a set of CSV files that describe the routes of the public transport service and the timetables over the routes. More in details, the files are:

- agency.txt: basic information on the transport supplier.
- stops.txt: geocoded list of stops.
- routes.txt: labels and description of the routes.
- trips.txt: identifiers of the various trips 9services) that are operated on a given route.
- stop_times.txt: for each trip_id, the arrival and departure times at one of the stops.
- calendar.txt: the days the services are executed.
- calendar_dates.txt: exceptions to the calendar above, such as holidays.

There are supplemental optional files containing information on fares, on how to display the routes on a map, on how to make transfers from a route to another, etc.

2.3 Modelling car pooling services

In SocialCar we also represent car pooling services, which are a hybrid transport mode halfway between public transport lines (car poolers driving to work tend to stick to fixed schedules) and private transport (car poolers are not bound to a specific route, they can change dynamically in front of varying traffic conditions).

For the above reason, car pooling services can be represented as bus services, in order to represent regular, planned services, while deviations will be represented using the data model proposed by GTFS-realtime[5].

The basic static information of a car pooling service, in order to consider it for planning transit routes are the same used to represent a public transit service in GTFS:

- agency.txt is extended with information on driver reputation.
- stops.txt are the current planned stops in the car pooling service. If there are no stops, these are the major waypoints. Here, a waypoint is defined as a specific place (interception, point of interest etc.) in the proximity of which the driver will pass by.
- routes.txt are not relevant, as usually a car pooler drives only two routes per day (from home to work and back), but some car poolers might travel multiple routes (e.g. from home to work, and then from work to another work location).
- trips.txt contains the trips associated with routes. While a bus travels the same routes many times per day, a car pooler will usually travel a route only once per day.
- stop_times.txt contains the expected time of arrival and departure at each stop along the route.
- calendar.txt contains the information on which days the route is travelled.

GTFS-realtime contains the essential information that is needed to describe alterations to planned routes. As described in the specification, the following information is supported:

- updates - delays, cancellations, changed routes.

[5] https://developers.google.com/transit/gtfs-realtime/

- alerts - stop moved, unforeseen events affecting a station, route or the entire network.
- current positions - information about where the vehicles are.

The possibility of notifying users about stops (pick up points) that have moved, as well as deviations in time from the expected schedule, is essential and very important for SocialCar.

Another essential element is the availability of places on a car (residual capacity). Unlike public transportation, cars have very limited capacity, and as soon as all places are taken, the car is no longer available as a service. The GTFS-realtime format must be then enriched with another element:

- residual capacity – the amount of available places on a car on a route segment.

In order to represent this information, the information on the car pooling service must be organised according to the segments which connect pick-ups and drop-offs. In Figure 6 a typical situation is represented. The car starts the trip at home with the driver and one passenger. At the next Stop, 2 passengers are picked up and then the car is fully occupied, meaning that there is no residual capacity to serve other passenger. This information must be communicated to the SocialCar system so that that this particular car pooling service is not chosen in a potential trip planning solution by the algorithm.

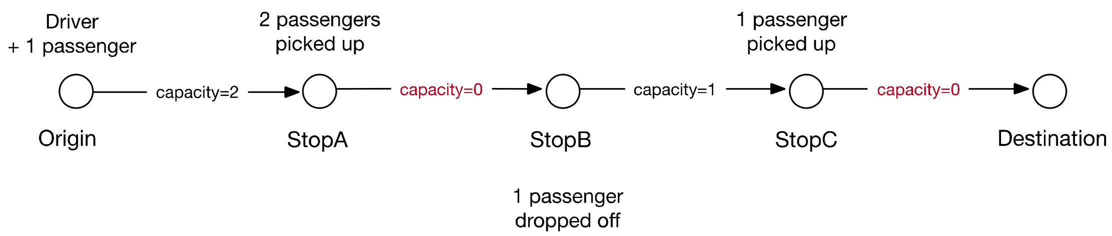


Figure 6. A car-pooling service with pick-ups and drop-offs.

2.4 Representing dynamic network information

The status of the traffic conditions in an urban area can greatly affect the solution of a routing problem. Finding the shortest path depends heavily on the traffic congestion level and on traffic disruptions.

In case of public transit, the information can be conveyed through the GTFS-realtime standard. As discussed in the previous section, GTFS-realtime can be used to share information about delays, cancellations, events affecting routes or stations, and also the vehicle positions, which are fundamental to accurately plan an intermodal route.

The online information to be made available for planning private transport is:

- Road blocks.
- Road works.
- Incidents and accidents.
- Road infrastructure status.
- Traffic related information (speed, flow, occupancy).

Actually, the only information that will be absolutely necessary in order for the SocialCar route planning algorithms to work efficiently will be the average travel time on road segments, which can be derived from the above listed traffic related measurement.

Traffic Message Channel is the most widely used information channel on live traffic information. Data transmitted over this channel is routinely integrated in commercial navigation systems.

The Datex II standard[6] is a EU supported standard for exchanging real time information on traffic conditions. Live traffic information is usually managed at a local scale by the Traffic Management System (TMS) of a given city.

Another source of traffic information at a global level, is provided by the GoogleMaps JavaScript API. GoogleMaps algorithms return distance matrices (shortest paths between origins and destinations) taking into account live traffic conditions, estimated from the average travel time returned by Android devices which consented to sharing GPS position data with Google.

2.5 Modelling demand: transport requests

On the demand side, the user wants to find a transport solution from origin to destination. The essential data to be used to feed the multi-modal planning algorithms are:

- User id: unique identifier for the user. Not strictly necessary for the algorithm, but it is needed to keep track of different user requests.
- Current location and current time: this information is required in case that the user needs to re-plan a trip.
- Origin of trip: it can be the current position or any pre-determined POI chosen by the user.
- Destination: the place where the user wants to get to. It can be any point on the map, or a pre-defined POI.
- Time of departure: the time at which the user is available to depart. Alternatively, it could be the arrival time i.e. the time the user wants to arrive at destination.
- Travel preferences, including: maximum number of modal changes, maximum number of intermediate stops, maximum amount of time spent waiting, maximum travel time, minimum reputation level of the driver.
- Another important parameter is the tolerance to variation in travel duration. This parameter is important in case a car pooling service needs to be re-routed. As most users will be by default not willing to compromise on travel duration and length, this should be incentivised by a reward mechanism.

[6] http://datex2.eu

3 Algorithms overview

In this section we make a brief analysis of considered algorithms for multimodal route planning, for ride matching, for tracking users, and for managing their reputation evolution. In Figure 7 we describe the overall interplay of the SocialCar algorithms.

The user can submit a transport request which is handled by the route planning algorithm, based on the techniques and methods described in Section 3.1. The algorithm plans the optimal route according to user preferences, taking into account the available transport options. The algorithm plans on the basis of available information, ideally also the online status of private and public transport networks. If available, the current position of car pooling offers is also considered.

The algorithm returns a number of route alternatives, which may involve some car pooling services. In those cases, a ride matching algorithm is invoked. Its task is to distribute potential concurrent requests for seats on car pooling vehicles and if needed, to dynamically assign travellers to car poolers, adding stops on a car pooling route. If a match is found, the car pooling offer needs to be accordingly updated (less seats on a vehicle).

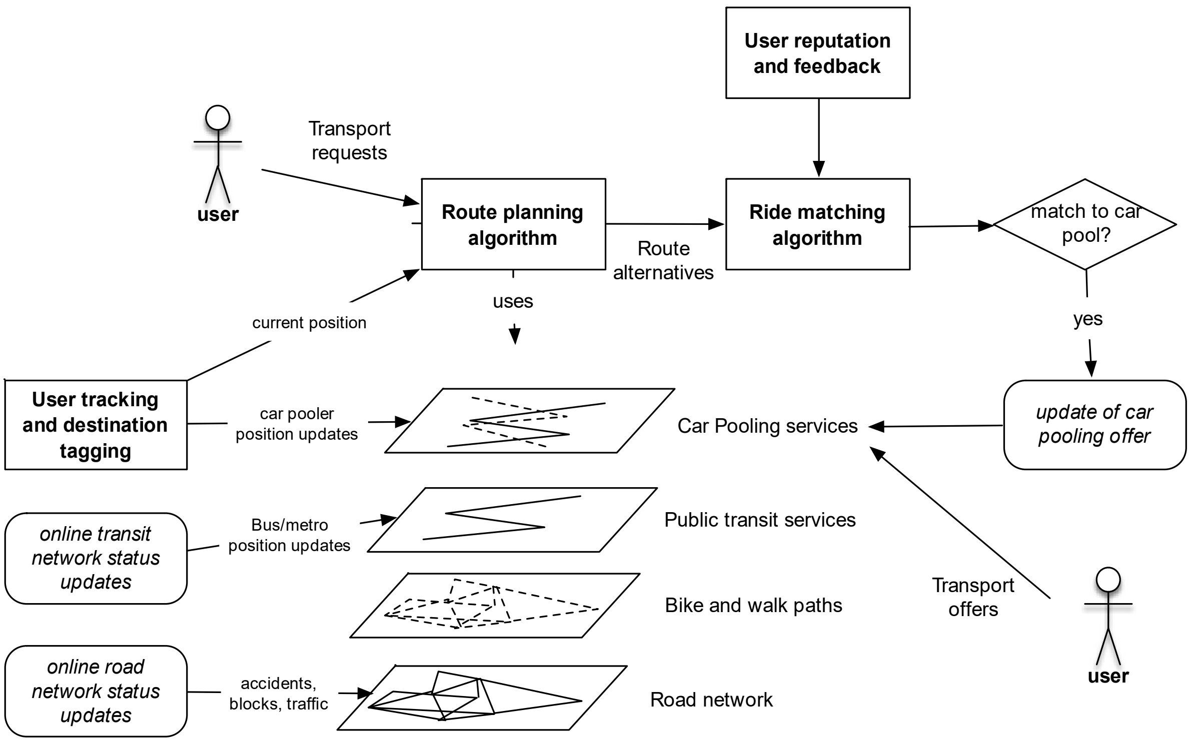


Figure 7. A general overview of the interplay of the SocialCar algorithms.

3.1 Algorithms for multimodal route planning

An overview over existing algorithms in the area of shortest path computation, specialized to multimodal route planning, is provided. The basis of many state-of-the-art algorithms forms *Dijkstra's Algorithm* (Dijkstra 1959), presented in 1959. This method computes a shortest path from a given source node to all other nodes, given an underlying graph representing, in our case, a road network. More specifically for road networks, there exist many algorithms to reduce the time of calculating a shortest path from one location to another. See, for example (Hart et al. 1968), that allows the usage of optimality preserving heuristics to decrease the number of iterations necessary to compute the distance to a given selected target node. One of the heuristics used in such a contexts are discussed in the Proceedings of SODA (2005), Möhring et al. (2005), and Bast et al. (2007). A substantial improvement in computing shortest paths on road networks has been provided in Geisberger et al. (2008).

With respect to modelling, transit networks, where connection times play an important role, are more complicated than simple road networks. For transit networks several models exist. The most common ones, according to Pyrga et al. (2007) are *time-expanded* and *time-dependent* networks. When the time-expanded model is considered, and when the underlying graph is manipulated appropriately, the algorithms developed for road networks can be applied directly. To have approaches very reactive to queries, several methods exist. A speedup idea is introduced in Bast et al. (2010). It is based on precomputing information at bootstrap of the system. Such a method leads to query times of a few milliseconds, also for extremely dense city-networks. More recently, a round-based approach that does not rely on precomputation has been introduced in Delling et al. (2012), although the efficiency is slightly worse.

In terms of optimization, multi-criteria optimisation is receiving growing attention nowadays. Among considered criteria there are total duration, car duration, walking duration, the number of transfers, costs, and when we consider the SocialCar framework, reputation information and adhoc rewards can be easily plugged in. Without a proper tuning, mapping multiple criteria linearly to one criterion can generate problems (Fleming et al. 2003), since some desired solutions are likely to be hidden away. Other approaches rely on *Pareto Sets* (Hansen, 1979), but tend to have too long computation times for a reactive environment like the one of SocialCar.

Modern approaches for transit networks (like Bast et al. (2008)) consider walking to nearby stations. Others consider different modes of transportation, restricted to a certain hierarchy (Yu and Lu, 2012). Less restricting multi-modal methods have recently been considered in Braun (2012) and Delling et al. (2012). The latter two papers show that using Pareto Sets with multiple criteria enlarges the set of optimal paths to an impractical extent, especially when car trip duration is considered (making these approaches impractical for the SocialCar characteristics). Interesting studies with academic overviews of the current state of the art can be found in Pajor (2009), Brodesser (2013) and Liu (2010).

Modelling reliability and robustness (i.e., how good are the alternatives, in case a transfer is missed) is another important aspect, not very much treated in the literature, as far as we are aware. Only recently, the work of Strasser (2012) focused on this topic. However, it is limited to transit networks. Older studies on robust and stochastic methods for different road network optimization can be retrieved in Montemanni and Gambardella (2004 and 2005), Montemanni et al (2008 and 2007). These methods have potential to be adapted to the proper transit networks required in SocialCar.

3.2 Algorithms for carpool matching

Many studies have been proposed to address inefficiencies in the traffic system, and car pooling has been identified as an effective means of indirect traffic management, since if it is implemented effectively, it leads to the reduction of the number of vehicles participating in the transportation system. Car pooling can be described as the cooperation of two or more persons regarding the use of a single vehicle to meet their personal needs. In addition to the social benefits mentioned before, there are potential personal benefits for the individuals taking part into car pooling. These benefits range from reduced fuel costs, to reduced toll fees, reduced time spent on the road, etc. A detailed overview on the history of car pooling and how solutions have changed during the years, also based on available technology, can be found in Xia (2015).

Modern carpool services use network-based algorithms to suggest a carpool team and commute route. The matching algorithms available are mainly designed for a single driver and a single passenger, or system-wide carpool matching optimization. In more details, genetic-based algorithms are discussed in Jiau (2013), Huang et al. (2014) and (2015), with fuzzy controls introduced in the latter. A study based on simulation is presented in Agatz et al. 2011. The main limitation of these methods is that they might not provide an optimized solution when a commuter wants to carpool with more than one person. It has been observed in Small et al (2006) and Poole and Balaker (2005) that two-person car pooling might be not efficient enough to provide any real benefit, so more complex solutions are needed. Three-person car pooling is discussed in Xia et al (2015), with potentials for generalization to larger groups.

3.3 Route planning and carpool matching for the SocialCar framework

When moving to algorithms for the SocialCar framework, the main challenge is to develop a method integrating together the ideas described in Section 3.1 multimodal route planning and in Section 3.2 for carpool matching.

In general, we can consider two alternative approaches to matching passengers to cars during the ride matching process: in the first approach, the car does not detour from its original path, and passengers need to travel to the closes possible stop. As described in Figure 8, the car travels from origin to destination along a pre-determined optimal route. The passengers can be picked up at intermediate stops which coincide with public transport stops in order to favour modal changes.

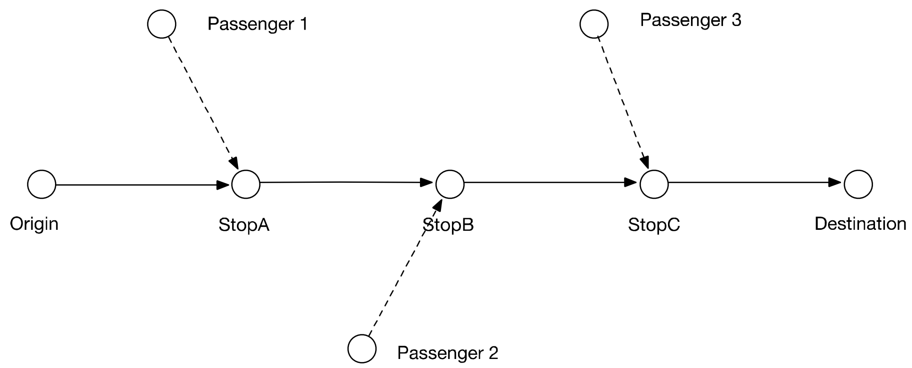


Figure 8. The car poolers are picked up at pre-determined stop points, along the optimal path from origin to destination.

There is the opposite approach, shown in Figure 9, of re-routing the car, in order to pick up passengers at their preferred location. This approach involves solving a routing problem, and it might be suboptimal for the car driver, as the expected travel duration can be considerably extended.

In SocialCar we will tend to privilege the first solution, as it strengthens the smile between car pooling services and public transport lines, but at the same time small detours will be allowed in order to retain the greater flexibility of car pooling over public transport.

A further challenge in the SocialCar context will be to plug reputation information about users and reward schemas into the optimization. This aspect should be overcome by considering these bits of information as weighted contributions to the optimization targets. In such a case, some tuning phase will be necessary to choose the right weights.

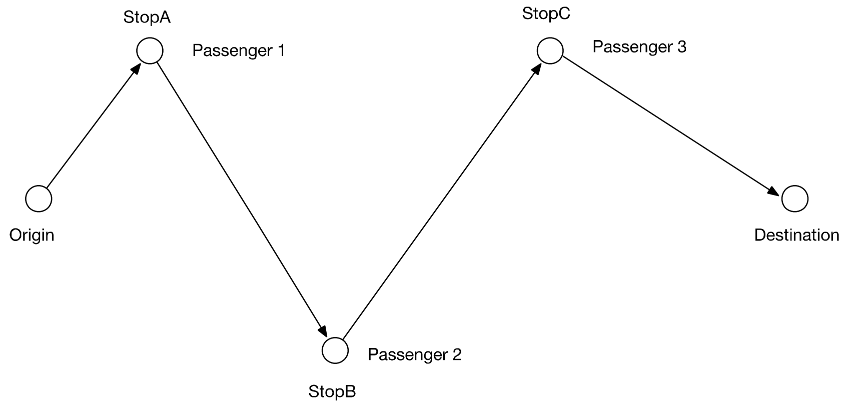


Figure 9. The car modifies the original route in order to pick up the passengers.

Concerning the algorithms for the SocialCar framework, the main challenge is to develop a method integrating together the ideas described in Sections 3.1 and 3.2 for carpool matching. This can be achieved by the use of algorithms of the Dijkstra family (Dijkstra 1959), eventually made more complex to take uncertainty into account. In particular, the main work will be in the preparation of the graph representing the network on which the algorithm itself will be run: with appropriate combinations of the layers described in Section 2.1, it will be possible to use the same approach both for route planning and carpool matching. Different versions of the method will be implemented and tested in order to find the best trade-off between speed and precision.

A further challenge in the SocialCar context will be to plug reputation information about users and reward schemas into the optimization. This aspect should be overcome by considering these bits of information as weighted contributions to the optimization targets. In such a case, some tuning phase will be necessary to choose the right weights. Again, such an approach can be easily fitted into methods of the Dijkstra family (Dijkstra 1959). Notice that using different set of weights can lead to alternative solutions, which is good because these different solutions can be proposed as alternatives to the decision makers, that select the one most suitable to their need/ideas.

3.4 User tracking and destination tagging algorithms

User tracking uses mobile phone GPS recordings to deliver the trajectories followed by users. These tracks can be then analysed for different purposes such as destination tagging, which is made possible through trajectory mining.

User tracking addresses a number of issues related to the raw positioning data; this is the pre-processing stage, where a number of techniques are applied, such as noise filtering, trajectory compression, trajectory segmentation and map matching (Zheng 2015). At this stage, it is also important to keep in mind that the disclosure of a user's trajectory can cause a privacy leak, which should be addressed through privacy preserving techniques.

Finally, the mining of the pre-processed trajectories allows the extraction of generic patterns which can summarize the dataset, such as moving together patterns, trajectory clustering, sequential patterns, or periodic patterns (Zheng 2015). Classification models allow to associate trajectories to specific activities (e.g.: running, hiking) or transportation modes (e.g.: car, bicycle).

3.4.1 User tracking

User tracking, when referred to geo-location, consists of successive position calculations performed at a high enough rates to obtain the user trajectory, whether stationary or moving. In the scope of SocialCar use cases, localization (i.e. the position calculations) may occur at the user smartphone via different positioning methods (satellite-based like GPS or Glonass, network based using cell-phone and/or Wi-Fi networks).

These methods provide position calculations with different degrees of accuracy, but none of them is exempt of possible important inaccuracies, mainly due to sensor noise, poor signal, multipath, own physical performance limitations of the techniques, and other possible factors. Therefore the mitigation of positioning errors is the main issue to be faced when aiming at obtaining accurate user trajectory estimations.

Whenever the noisy position estimates cannot be recovered through techniques such as map matching, it is necessary to filter them in order to remove them from the dataset. For that purpose we can use from simple methods based on mean (or median) or more sophisticated approaches, such as the Kalman and particle filters (Lee and Crumm, 2011) or heuristics-based outlier detection (Yuan et al 2010, Yuan et al 2011, Yuan et al 2013, Yuan et al 2009).

Considering that positioning sensors (like GPS receivers) are attached to mobile devices such as smartphones, battery consumption is an issue. For that reason performing position calculations all the time at high rate can be a wrong decision and alternative approaches like avoiding calculations in some time periods can be suitable. Recording coordinates with a high sampling rate (e.g.: every second) also produces considerable overhead, in terms of storage, processing and communication. To address this issue, there are compression techniques, aimed at storage efficiency; the underlying idea is to minimize the size of the trajectory, while preserving the accuracy of the representation. The main strategies to solve this problem are offline compression, which reduces the trajectory size after it has been generated, and online compression, which processes the trajectory as it is being generated (Zheng 2015).

Trajectory segmentation is a compulsory pre-processing step, required by algorithms such as clustering and classification. It basically divides the trajectory into segments, thus reducing their computational complexity. The criteria for this segmentation can be the time interval, the shape of the trajectory, or its semantic meaning. This last criteria is the base of a group of methods that identify key points (stay points), or transportation modes, in order to split the segments (Yuan et al. 2013, Lee et al. 2008, Zheng et al 2008 a and b).

Trajectory segmentation is also used for identification of sub-trajectories. This point will be further discussed later. (see description of TACLUS algorithm in 3.4.2.1).

Map matching algorithms assign sequences of coordinates to existing road segments. This allows to associate a trajectory to a particular infrastructure (e.g.: road), from which we may have other information (e.g.: condition, traffic, number of lanes). Map matching approaches can rely on additional information or on the range of sampling points, in order to execute this task. Additional information could be the geometry of the roads (Greenfeld, 2002) or their connectivity (Chen et al. 2003); more advanced methods can use probabilities to choose from multiple paths, whenever GPS noise establishes some degree of uncertainty (Ochieng et al. 2004, Pink and Hummel, 2008), or they may even combine more than one approach (Newson and Krumm, 2009). Range-of-Sampling-Points based algorithms can follow a greedy, local incremental, strategy, sequentially extending the solution from an already matched portion (Civilis et al. 2005), or alternatively they can follow a global approach, matching an entire trajectory with a road network (Brakatsouls, 2005). As global algorithms are more accurate, but less efficient than local-incremental methods, they are generally used in offline tasks. Finally, advanced algorithms can combine both methods (local and global) to approach problems where there is little information (e.g.: low GPS sampling rate) (Newson and Krumm, 2009).

3.4.1.1 Proposed approach

The proposed approach for user tracking employing the user smartphone considers the following issues as performance aspects to be addressed and improved:

- inaccurate user trajectory estimation due to noisy position estimates
- high power consumption
- high storage and communications consumption

In this section the proposed method for addressing each issue is explained. These methods will be implemented at the smartphone and will perform actions in real time for improving the trajectory estimation.

3.4.1.2 Inaccuracies on user trajectory estimation

A three layer approach to filter the noise of the position estimates is proposed, as a three steps processing chain for the raw positioning samples obtained from the positioning sensors at the smartphone. At the lowest level, and as a first processing step, a quite simple method for removing outliers will be applied. This step consists of discarding the positioning estimations with a positioning error that exceeds a certain threshold value. An estimation of the positioning error is provided by the smartphone OS for each delivered position sample, at least considering the most popular OS like Android and iOS. The threshold value can be adaptive to certain aspects such as the employed positioning technology in each case, so for example the value will be more restrictive for GPS-Glonass than for positions obtained with less accurate methods like the ones based on terrestrial communications networks.

The second step consists of applying a probabilistic tracking filter in order to obtain a smoother user trajectory estimate, thus reducing the positioning error and noise. This more complex approach gives more satisfying results than simple temporal averaging filters. The idea behind is not to consider each position calculation like a one shot sampling, but taking benefit from past estimated positions assuming a certain motion model that governs the displacement of the user. The general assumption is that the user does not perform erratic jumps when moving but follows a quite smooth trajectory. For each position calculation (epoch) a position prediction based on the motion model is performed, and then in a second stage of the filter the prediction is corrected with the observable information. In our case the observable data corresponds to the position estimate resulting from the first processing step described before.

A particle filter will be employed for the user tracking in SocialCar. Among the probabilistic tracking filters, particle filter is known to outperform other alternatives (like Extended Kalman filter) for geo-localization purposes. The price to pay is a higher computational requirement, but this is not a problem for its implementation in current COTS smartphones and tablets. The particle filter, also known as Monte Carlo simulation, represents the probability density function of the user position by a set of random samples (i.e. the particles) each one with an associated state (the position coordinates) and weight (the posterior probability considering the observables). Each particle explores the environment according to the defined motion model for the mobile target (i.e. the user), and its weight is updated at the prediction stage. The position estimate is obtained from the probability density function that the set of particles make up.

A proper number of particles will be chosen for a proper execution of the filter at the smartphone, considering a trade-off between positioning performance and computational cost. In order to be adaptive to the wide range of possible motion models resulting from the different possible transport modes to be supported, a dynamic prediction stage that obtains the user acceleration and heading magnitudes from the smartphone inertial sensors (accelerometer, gyroscope, magnetometer) will be explored. The intention is to achieve a filter that adapts as much as possible to the transport mode employed by the user in each case, in order to reach the best possible trajectory estimation whichever the situation is.

The third step of the position samples processing consists of map matching making use of the available geographic routes that can be obtained from the map information. This step has two main objectives, which are contributing to the trajectory accuracy improvement and ease the later positions management. Depending on the finally available map information, a global or local incremental approach will be followed. The fusion of this step with the particle filter will be explored, in order to improve the efficiency of the solution; in that case the areas to be explored by the filter particles would be constrained to the routes taken from maps.

Figure 10 illustrates the proposed three steps processing chain that will be performed in real time at the smartphone after the positions sampling.

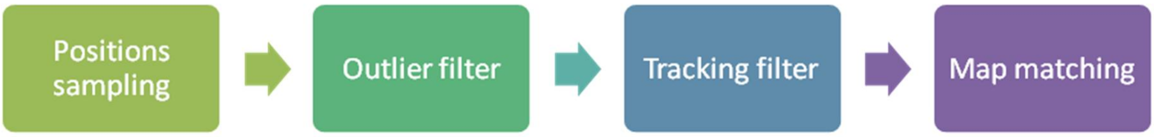


Figure 10. Processing steps for mitigating the positioning errors in user trajectory estimation.

3.4.1.3 High battery consumption

Continuous position calculations, especially using satellite-based methods like GPS or Glonass, would drain the smartphone battery in few hours. For that reason we propose the combination of three mechanisms in parallel to mitigate as much as possible the power consumption while preserving the quality of the user tracking.

The first mechanism consists of avoiding the position calculations when the user is not moving, because in the absence of displacement it brings nothing new to calculate new positions to update the estimated trajectory. To this end it is necessary a method to identify when the user is still, i.e. is not provoking any variation in its geographical location. The proposed method to achieve this will make use of the smartphone inertial sensors (mainly accelerometer and gyroscope), performing an analysis of the three axis acceleration and angular rates magnitudes. In order to achieve a method robust to the different transport modes, a mechanism to differentiate between a walking action and using a vehicle will be explored. The transport mean differentiation should be done by generating models that map the device's generated information (i.e., GPS and other sensors, such as accelerometer and gyroscope) to the actual mean. These models could be built using laboratory-controlled experiments to generate the ground truth. We propose to run several experiments and automatically generate such models using Machine Learning based techniques (I.e., supervised classification). Many different existing algorithms could be tested.

The second mechanism consists of a dynamic adaptation of the positioning sampling rate, depending of different possible aspects like specific use case requirements or transport mode.

The third mechanism consists of, given a user tracking request, selecting the optimum positioning technique (i.e. position provider at the smartphone) taking into account that the one able to satisfy the accuracy requirements at the minimum consumption cost must be selected. This way, the provider that makes use of terrestrial communications network (3G-4G, Wi-Fi) will be preferred to satellite-based (GPS, Glonass) whenever they fit the performance requirements. This aspect depends on the mobile OS geo-positioning features.

Modern operating systems (iOS, Android etc.), in fact, already provide APIs for geo-positioning, making the information source (e.g., GPS, Assisted GPS, wi-fi, etc.) transparent for the developers, without caring of optimization such as trade-off between battery consumption and accuracy.

3.4.1.4 High storage and communications consumption

In order to reduce the storage, processing and communications overhead that results from recording geographical positions for tracking purposes, a compression technique will be applied at the smartphone as part of the user tracking software.

3.4.2 Trajectory Analytics

Trajectory analytics include descriptive techniques such as pattern mining, as well as predictive approaches such as classification. Stay point detection, is a technique which amplifies the information of a tagged location, with some semantic content, and it can serve as a basis for location tagging.

Moving together patterns is a set of algorithms that aims to detect groups of objects moving together for a certain period, such as flock (Gudmundsson and Kreveld, 2006) (Gudmundsson et al. 2004), a convoy (Jeung et al. 2008 a and b), a swarm (Li et al. 2010), a traveling companion (Tang et al. 2012 a and b), or a gathering (Su et al. 2013). They tend to use a density-based approach to find a cluster of moving objects, although the distance metric may be extended with other factors (e.g.: semantic factors, speed) (Jensen et al 2007).

A common approach to cluster trajectories, is to represent a trajectory with a feature vector and to denote similarity between trajectories through the distance from their feature vectors. The generation of comparable feature vectors, and the encoding of the properties of the trajectory in this entities is not, however, a trivial problem (Zheng 2015). The regression mixture model combined with the Expectation-Maximization (EM) algorithm, groups similar trajectories (Gaffney and Smyth, 1999) (Lee et al. 2007) and partitions them into line segments, building close groups of segments, using the Trajectory-Hausdorff Distance. For the problem of clustering data streams (e.g.: real-time clustering), it has been proposed the Micro-and-Macroclustering framework (Lee et al. 2007), which finds first micro-clusters of trajectory segments, and then groups them into, high-level, macro-clusters.

Sequential patterns can be defined as a group of objects travelling together a common sequence of locations in a similar time interval. This differs from moving together patterns, in the sense that in sequential patterns, objects do not have to be travelling at the same time, and they may just share a common sequence. The identification of these patterns has been used for applications such as travel recommendations, life pattern understanding, next location prediction and estimating user similarity (Zheng 2015). Due to its characteristics, this problem can be sub-divided into sequential pattern mining in the free space and sequential pattern mining in a road network (with map matching).

Periodic patterns can provide an insight into activity behaviours (e.g.: commuting), and help to predict future movements. Due to the fuzziness of spatial locations, time series methods are not directly applicable to trajectories (Zheng, 2015). The frequent pattern mining minimum threshold concept, has been the basis of an algorithm to retrieve maximal periodic patterns from trajectories (Cao et al. 2007). The two-stage detection method, based on clustering and time series methods (e.g.: autocorrelation, Fourier transform) is another approach to this problem.

Trajectory classification relies on attributes such as transportation modes or human activities, to differentiate trajectories, or part of them. Trajectory tagging focus on attributing a semantic label to this trajectories, based on those attributes; it provides the basis for intelligent methods such as recommenders and context-aware computing (e.g.: Google Now). Before building a classification model, it is necessary to pre-process the trajectories, by dividing them into minimum inference units (segments or points), and extracting features from those units. This pre-processing relies on techniques discussed on the "pre-processing" section, such as trajectory segmentation and map matching. The classifications models can range from decision trees to sequence inference models or Markov chains, and they have been applied in different contexts. For example, one model classified the mobility of an user (e.: stationary, walking, driving), based on GSM signals (Timothy et al. 2006). The transportation mode has been the focus of other models, for instance for distinguishing between driving, biking, bus and walking (Zheng, 2015). Another model focused on classifying the status of a taxi (e.g.: occupied, non-occupied, parked), based on the GPS trajectories (Zhu et al. 2011).

Finally, another method relevant for semantic tagging, and more specifically, destination tagging, is "stay point detection". This approach assumes that not all points in the trajectory are equally important, as people are more likely to stay for a while near spatial points of interest, such as tourist landmarks or shopping malls. There are two types of stay points: those where the user stays stationary for a while (and these are very rare), and those where people move around in small trajectories (near-stationary). Through destination tagging, we can attribute a semantic meaning to these stay-points, and create meaningful insights that can provide a basis for recommender systems (Zheng and Xie 2011), destination prediction (Ye et al. 2009), or gas consumption estimation (Xue et al 2013). Stay point algorithms rely on metrics of the distance and time span, and they have been improved using density based clustering concepts (Yuan, 2011) (Yuan 2013).

It has been demonstrated that a trajectory alone, may serve the purpose of identifying an individual, and extract information regarding her/his habits (Gruteser and Grunval, 2003). Apart from the ethical implications, there are some legal issues arising from this situation, as it breaks what is established in article 11 of the Data Protection Act (Paniza-Fullana 2010). To protect individual privacy, there are techniques that obfuscate the original information, by deliberately blurring it, while still ensuring some degree of utility on the trajectory data. On the case of historical trajectories, these techniques include clustering-based (Abul et al. 2008), generalization-based (Nergiz et al. 2009), suppression-based (Terrovitis and Mamoulis, 2008), and grid-based (Gid'ofalvi et al. 2007) approaches.

3.4.2.1 Proposed approach

We propose to develop a sub-trajectory identification. Given the specific use-case of Social Car, we don't plan to obtain the exact matching of trajectories (from source to destination). The system should rather search for common sub-trajectories among users in order to proactively suggest travelling peers. There is already literature on this (in particular TACLUS algorithm published in SIGMOD07 by JG Lee et al.). TACLUS works in two steps:

1. Trajectory segmentation. As discussed before, it's important to make the right assumptions about the segmentation degree. Small segments allow for a finer-grained clustering, but imply higher computational cost (mostly for the back-end). Bigger segments offer a coarse view of the trajectories.
2. Segment clustering. The segments are represented as vectors and clusterized with the well-known DBScan algorithm (already present in many libraries for different programming languages).

After the sub-trajectory identification implemented by the algorithm based on JG Lee et al we propose to work with the matching of complete trajectories, from the origin of Social Car users and the destination tagging, in order to inform the system about common trajectories. The idea of working with sub-trajectory instead of full-trajectories is that the system could proactively suggest peers. In the example of Figure 11 user A will pass by, in turn, homeB and workB, where <homeB,workB> is a subset of <homeA,workA>. Hence, the system could suggest to A and B to share the car (owned by A).

Figure 11. Infering peers from trajectories of users.

We propose to identify the nature of these trajectories (home-work, work-home, home-leisure, etc.), taking into account the ethical issues mentioned before. If user can manually flag the trajectory (origin position and destination) it will be much easy, but is possible to identify them according to the frequency, schedule, etc.

3.5 User reputation algorithms

The assessment of the reputation of users is considered to be a key factor when it comes to the design, manufacturing and operational process of every system that relies on random user information. Systems of this type are primarily functioning online, covering various activities such as e-commerce, online auctions, questioning and answering and so forth. In those systems a reliability metric (reputation) value for each user is usually calculated by a mathematical model, which is based mainly on rating of other users on the information or services that a user in question provides. There are some cases though, in which users may give unreasonable or unpredictable ratings, because either they are not familiar with the rating process, or the related field. Another case is the one involving malicious users who are willing to inject false information into the system in purpose. A malicious user may give high ratings to low quality information (or vice versa) or perform a so-called Sybil attack, by creating a large number of pseudonymous entities, and using them to gain a disproportionately large influence (Lazzari, 2010). A reputation system's vulnerability to a Sybil attack depends on how cheaply user accounts can be generated, the degree to which the reputation system accepts input from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically. Therefore various methods and mechanisms have been proposed for effectively filtering user ratings for user reputation evaluation.

In the field of e-commerce and online auctions, the reputation of a seller or a buyer depends on the quality of the object that sells or buys, respectively. In turn, the quality of an online object depends on user ratings. These ratings are generated by users who may have bought these objects and were happy (or not) based on quality-related criteria, or by malicious users that just provide arbitrary ratings for unknown reasons. In this context, most of the rating systems iteratively calculate the quality of online objects and then user reputation. Liao et al. (2014) have proposed such an iterative method, which enhances the influence of highly reputed users in modulating the quality of online objects, while penalizing the corresponding of the malicious ones, leading to more accurate object quality evaluation. Also work in (Zacharia and Maes, 2000) proposes an iterative machine learning method for estimation of user reputation in both loosely and strongly connected marketplaces, which can also be used in other online communities, such as newsgroups and mailing lists. Moreover, fuzzy based algorithms, which can better handle uncertainty, fuzziness and incomplete information in trust reports provided by sellers and buyers have been proposed (S. Song et al., 2005). Finally, the correlations between sellers and buyers have been employed for the design of reputation estimation algorithms such as in the case of (Morzy, 2008) where two reputation rating estimation algorithms are proposed that utilize the correlations between users. The first one calculates the credibility (reliability metric) of users by an iterative search of interparticipant connections, whereas the second one discovers clusters of users who are densely connected through committed auctions.

Apart from e-commerce and online auctions there exist various online activities, for which reputation estimation mechanisms have been proposed. Bian et al. (2009) deal with the problem of users' reliability estimation in the Community Questioning Answering (CQA) field. In particular the authors propose a semi-supervised coupled mutual reinforcement framework for simultaneously calculating the quality and reputation of questions and answers posted by the users (Jiang Bian et al., 2009). Also, (Wu et al, 2014) presents a method for trust level evaluation of cloud services, which facilitates the recognition and filtering of unfair ratings. Finally, (Zheng et al., 2008) proposes a collaborative framework for Web services reliability assessment for users from different geography locations.

An important and commonly used example of successful reputation management is the online auction website eBay, whose feedback mechanism supports buyers to build trust to unknown sellers and elicits honest behaviour (Resnick et al., 2002). In eBay's reputation system, buyers and sellers can rate each other after each transaction, and the overall reputation of a participant is calculated as the sum of these ratings over the last 6 months. This well-established reputation system adopted by eBay relies heavily on a centralized architecture to store and manage the ratings and it has been already implemented in some ridesharing matching agencies (see e.g., Avego, Carpool World, and Golco).

In academia, eBay-like reputation systems are well-studied and the most critical issue about them is how to ensure that it is in the best interest of a rational user to actually report reputation information truthfully (Jurca and Faltings, 2003). It has been found that:

- Reporting positive feedback can lead to increased competition with others in the future (Jurca and Faltings, 2003), (Dellarocas, 2006).
- Fake negative feedback can cause scarce resources to exit the competition (Jurca and Faltings, 2003), (Dellarocas, 2006).
- There is a bias towards a positive report in order to protect against a retaliation of a negative report by a counter partner (Resnick and Zeckhauser, 2002).
- It is easy to create a new ID to wipe out past records (Dellarocas, 2006), (Jøsang et al., 2007), (Witkowski et al., 2011).
- All service providers are not around long enough to be incentivized by future returns that depend on today's feedback (Dellarocas, 2006), (Jøsang et al., 2007), (Witkowski et al., 2011).

In the context of SocialCar, the application asks both drivers and passengers to rate each other, but only after the end of every shared ride (as in eBay system only after each transaction). The reputation gained through the rating process is an influent factor, which is considered in the decision on whether car pooling with another person or not. Such scores are also used in future trips so that seats offered by car drivers with high reputation appear in better positions than others with lower reputation. For this purpose, a reputation algorithm should be provided with a suitable data model for user rating in order to effectively calculate the credibility (reliability metric) of ride participants. The proposed data model for user rating, based on Deliverable D1.1-"The SocialCar Arena", includes these *subjective properties*:

- Punctuality (it could be also an objective measure)
- Cleanliness
- Courtesy
- Facilities (Air Conditioning)

- Luggage capacity
- Pets allowed
- Driving style

Moreover, some other *objective properties* are calculated or derived by the system, such as:

- Punctuality
- Ride acceptance
- Member Seniority
- Age
- Driving style (average speed)
- Travelling time

In order for the reputation algorithm to efficiently calculate a result, each one of the above properties is converted into an integer valued from 1 to 5. In case of a categorical property, such as Ride Acceptance, a class repartition is performed and every class is assigned to a corresponding value (e.g. very low -> 1, low -> 2, regular -> 3, high -> 4, very high -> 5). Finally, in case of a Boolean property, such as Pets allowed, a value of 'true' is assigned to 5 while a value of 'false' is assigned to 1.

The data model for user rating proposed above is a satisfactory set, in quality and quantity, of all the properties that should be considered as potential inputs to the SocialCar user reputation algorithm. However, the final selection of the particular properties on which a user should be rated depends on the role of the user during a shared trip. In the context of SocialCar, there exist two main user roles as explained in Deliverable 1.1:

- Driver
- Passenger

When it comes to user evaluation, a driver can rate the passengers, whereas a passenger can rate the driver, but also a travel mate. In this context, a data model for *driver rating* should include all the aforementioned properties, but a passenger rating model should be a subset of the above, including only the properties that make sense for a passenger. In particular, a data model for *passenger rating* should include these *subjective properties*:

- Punctuality
- Courtesy

and these *objective properties* derived by the system:

- Punctuality
- Member Seniority
- Age

The proposed reputation algorithm is a direct derivation and adaptation of existing reputation algorithms. By using it, users can rely on a quantitative measure to decide whether trusting another user or not. The total score (a real number from 0 to 1) that represents user's reputation is a weighted sum, where each factor stands for each one of the above properties (depending on user's role) and each weight (importance factor) depends on the importance of the corresponding property. In cases where a SocialCar user represents a passenger, the total score is calculated with all driver related property (e.g. driving style) weights set to zero. The score calculation can thus be derived from the following formula:

$$S = w_{pu}R_{pu} + w_{cl}R_{cl} + w_{co}R_{co} + w_{fa}R_{fa} + w_{lu}R_{lu} + w_{pe}R_{pe} + w_{dr}R_{dr} + w_{ri}R_{ri} + w_{me}R_{me} + w_{ag}R_{ag} + w_{tr}R_{tr}$$

As shown above, there are eleven different parameters that shape the total reputation score:

- R_{pu} : *Punctuality* of the user, meaning arriving exactly at the time appointed.
- R_{cl} : *Cleanliness* of the shared vehicle and therefore, the tidiness of the driver.
- R_{co} :*Courtesy*, or in other words, the politeness of the user during the trip.
- R_{lu} : Level of *facilities* (air condition, heated seats etc) that the shared vehicle affords, or the driver provides.
- R_{lu} : Luggage capacity of the shared vehicle.
- R_{pe} : This paramter is related to allowance of pets in the shared vehicle or not.
- R_{dr} : Driving style of the user (e.g. cautious, aggressive etc).
- Rri : Ride acceptance of a driver, reflecting how often a driver accepts a passenger for a ride when given the chance to do so.
- R_{me} : User's membership seniority.
- R_{ag} : The age of the user.
- R_{tr} : Total travel time of user, as recorded by the SocialCar system.

All the above factors represent average values collected from user ratings, system calculations or a combination of both. The reputation score that a user can get is normalised in the scale 0 to 1. Intermediate reputation values are translated into the five-star system as indicated below:

- [0 – 0.2) -> 1 star
- [0.2 – 0.4) -> 2 stars
- [0.4 – 0.6) -> 3 stars
- [0.6 – 0.8) -> 4 stars
- [0.8 – 1] -> 5 stars

3.5.1 Reward mechanisms

The main challenge of a car pooling system in order to be functional is the need to reach a critical mass of users. Without wide adoption of it, potential users will not able to find any partners, thus losing interest about the application and probably switching to a different one. For this purpose, user rewards and incentive mechanisms are a key factor when it comes to attracting users in order to increase membership.

Car pooling has been found to have a strong impact on the mode of commuting at the workplace level, attracting up to 15% of trips by only offering information, incentives and a partner matching program. Financial incentives, such as reduced parking costs or fuel subsidies, can drive that share higher (York and Fabricatore 2001). Car poolers tend to come from higher income, white-collar households and tend themselves to reside in areas not generally well served by public transport (Winters et al., 1991). Workplaces that are not served well by public transport are considered good candidates for introducing car pooling schemes.

Car pooling incentive mechanisms may incorporate a variety of means to encourage users to carpool. Possible incentives include reduced cost or free parking, preferred parking, or reward programs (such as prize drawings). In most of the cases, incentives should be based on particular circumstances that may motivate the users to carpool. For example, a user located in a downtown location with expensive parking might consider reduced cost or free parking as a good incentive, while a suburban employer might rely on prize drawing to be motivated.

Systems that make use of rewarding mechanisms are presented below:

Users of *Foursquare* can earn badges for completing certain actions through the site (see Figure 12). For example, the first time someone checks in, they are awarded the 'Newbie' badge. Once they have checked in to a total of 10 venues, they are instead awarded the 'Adventurer' badge. When the user has checked into 25 venues, the user gains the 'Explorer' badge and, when this figure reaches 50, the 'Superstar' badge is obtained. These badges are displayed on the user's profile. This gamification should encourage members to user their account more so that they can obtain more badges.

	Adventurer	You've checked into 10 different venues!
	Explorer	You've checked into 25 different venues!
	Superstar	You've checked into 50 different venues!
	Bender	That's 4+ nights in a row for you!
	Crunked	That's 4+ stops in one night for you!
	Local	You've been at the same place 3x in one week!
	Super User	That's 30 checkins in a month for you!
	Player Please!	Look at you, checking-in with 3 members of the opposite sex! Player, please!
	School Night	Checking-in after 3am on a school night? Well done!

Figure 12. Some of the badges available to earn on Foursquare with a description of how they are earned.

Similarly to Foursquare, *Stack Overflow* offers its members badges for completing certain actions or if their question proves particularly popular, well viewed or helpful. Multiple badges are obtainable. The easier to gain ones are ranked as Bronze level badges, with the Silver and Gold ranked ones proving less frequently obtained. The badges fall under multiple categories, depending on what they relate to. As an example, the list of badges that a member can earn because of questions they have asked are shown in Figure 13. Members can also gain awards for answering questions, participating more generally and moderating other behaviour.

Question Badges

Altruist	First bounty you manually award on another person's question	6k awarded
Benefactor	First bounty you manually award on your own question	28.7k awarded
Curious	Ask a well-received question on 5 separate days, and maintain a positive question record	201k awarded
Inquisitive	Ask a well-received question on 30 separate days, and maintain a positive question record	20.2k awarded
Socratic	Ask a well-received question on 100 separate days, and maintain a positive question record	2.3k awarded
Favorite Question	Question favorited by 25 users	26k awarded
Stellar Question	Question favorited by 100 users	3.5k awarded
Investor	First bounty you offer on another person's question	11.9k awarded
Nice Question	Question score of 10 or more	312.2k awarded
Good Question	Question score of 25 or more	91.1k awarded
Great Question	Question score of 100 or more	13.4k awarded
Popular Question	Question with 1,000 views	2.4m awarded
Notable Question	Question with 2,500 views	1.2m awarded
Famous Question	Question with 10,000 views	293.2k awarded
Promoter	First bounty you offer on your own question	50.6k awarded
Scholar	Ask a question and accept an answer	1.1m awarded
Student	First question with score of 1 or more	1.3m awarded
Tumbleweed	Asked a question with zero score, no answers, no comments, and low views for a week	649.3k awarded

Figure 13: Some of the badges available to earn on Stack Overflow with a description of how they are earned.

BlaBlaCar awards its members an experience level ranking, ranging from 'Newcomer' when the member first joins, to 'Ambassador' (see Figure 14). The higher a user's experience level, the more trustworthy a member is perceived to be by others. The level is based on four factors: email and phone verification, profile completion, the number of ratings from other members and the percentage of these that are positive and how long the user has been a member for. Therefore, users can progress their experience level by adding their bio, photo and their car details in addition to their name and journey information. Their level will also increase as other users give them good feedback ratings.

In the context of SocialCar, offering incentives is an effective way to increase membership and remove any preconceived barriers that the users may have. Incentives may include:

- *Priority parking*: By proactively taking into account parking spaces at regions of high car pooling demand, SocialCar system can create an effective user incentive. Priority parks are usually the parking spaces closest to these high demand regions or in the nearest parking building, and are a huge incentive where the parking supply is limited or where drivers have to walk long distances between the car park and their actuall destination. Parking permits should be used to identify the priority spaces, which also helps to raise the visibility of SocialCar carpool scheme.

- *Financial incentives*: These can include cash rewards, prize drawings or rewards in the form of free or discounted parking fees. Financial rewards may also include free or discounted tickets for the SocialCar application. Setting up a carpool 'miles scheme' can also be effective, which awards participants with gift vouchers or discounts at local restaurants, or shops for the number of miles they travel in a carpool.

In order for a user to redeem one of the possible rewards described above, SocialCar application should adopt a Point System strategy. A user can earn points every time another user rates him/her and the number of points earned is proportional to the rating received (e.g. 5 stars equal to 5 points). At the end of every week, a user can redeem his/her points in order to get a reward whose value depends on the total number of points spent. If more than one reward is available, then the user can choose the desired one from a list of all the available rewards provided that his /her points are adequate.

	Newcomer	Intermediate	Experienced	Expert	Ambassador
Verified email and phone	Welcome!	✔✔	✔✔	✔✔	✔✔
Profile completion		> 60%	> 70%	> 80%	> 90%
# of positive ratings received		1 rating	3 ratings	6 ratings	12 ratings
% of positive ratings received		>60%	>70%	>80%	>90%
Seniority		1 month	3 months	6 months	12 months

Figure 14: The different experience levels that a BlaBlaCar member can be awarded and how each ranking is achieved.

4 Conclusions

In this report we have made a preliminary analysis of the current state of the art in algorithms for multi-modal route planning. Our analysis has highlighted that a time-dependent representation of the road and transit network is suitable to be used for SocialCar as far as trips are planned within the urban region. Long range multi modal trip planning is not in the scope of the SocialCar project.

From this analysis it appears that the SocialCar core services can be based on Open Source data such as OpenStreetMap. Furthermore, tools such as OpenTripPlanner also provide a good starting base for the development of the route planning algorithms.

The GTFS format and its extension GTFS-realtime is also suited as a data model to represent car pooling services, as the main aim of the SocialCar project is to include car poolers in the range of supplied services on a transit network.

The algorithms for route planning have been reviewed, identifying the need for advanced versions of the traditional Dijkstra algorithm on static networks, adapted to dynamic conditions, where robustness to perturbation in travel times can play a role.

Finally, this report has reviewed how the demand side, that is user transport requests, can be modelled, and how user-oriented services, such as tracking his/her position using GPS services, can be used to improve the overall performance and user experience of SocialCar, as well as algorithms to monitor and build user reputation and user motivation by rewarding schemes.

This report is the first of a series of two. It lays the groundwork on which to develop an initial version of the SocialCar platform. The report highlights the research and development areas that will be pursued during the next stage of the project and a final version will be published at month 30 in the projects' life.

The output of this document will be now integrated in the design of the SocialCar platform by WP3 and the first implementations will be realised in WP4.

References

Abul O, Bonchi F and Nanni M (2008). Never walk alone: Uncertainty for anonymity in moving objects databases. In Proceedings of the 24th IEEE International Conference on Data Engineering. IEEE, 376–385.

Agatz NA, Erera AL, Savelsbergh MW, Wang X (2011). Dynamic ride-sharing: A simulation study in metro Atlanta. Transportation Research Part B: Methodological, 45(9), 1450–1464.

Bast H, Funke S, Matijevic D, Sanders P, Schultes D (2007). In transit to constant time shortest-path queries in road networks. In ALENEX.

Bast H, Carlsson E, Eigenwillig A, Geisberger R, Harrelson C, Raychev V, Viger F (2010). Fast routing in very large public transportation networks using transfer patterns. In ESA (1), 290–301.

Braun B (2012). Multi-modal route planning with transfer patterns. Master's thesis, University of Freiburg.

Brakatsouls S, Pfoser D, Salas R and Wenk C (2005). On map-matching vehicle tracking data. In Proceedings of the 31st International Conference on Very Large Data Bases. VLDB Endowment, 853–864.

Brodesser M (2013). Multi-modal route planning. Master's thesis, University of Freiburg.

Cao H, Mamoulis N and Cheung D. W (2007). Discovery of periodic patterns in spatiotemporal sequences. IEEE Transactions on Knowledge and Data Engineering 19, 4, 453–467.

Cadez I. V, Gaffney S and Smyth P (2000). A general probabilistic framework for clustering individuals and objects. In Proceedings of the 6th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.ACM, 140–149.

Chen W, M. Yu, Z. Li and Chen Y (2003). Integrated vehicle navigation system for urban applications. In Proceedings of the International Conference Global Navigation Satellite System. CGNS, 15–22.

Civilis A, Jensen C.S, Nenortaite J, and Pakalnis S (2005). Techniques for efficient road-network-based tracking of moving objects. IEEE Transactions on Knowledge and Date Engineering 17, 5 (2005), 698–711.

Gruteser M and Grunwal D (2003). Anonymous usage of location-based services through spatial and temporal cloaking, MobiSys '03 Proceedings of the 1st international conference on Mobile systems, applications and services, 31-42.

Dellarocas C (2006). Reputation mechanisms. In: Hendershott, T. (Ed.), Handbook on Information Systems and Economics. Elsevier, 629–659.

Delling D, Pajor T, Fonseca R, Werneck F (2012). Round-based public transit routing. In ALENEX, pages 130–140.

Delling D, Dibbelt J, Pajor T, Wagner D, Werneck RF (2012). Computing and evaluating multimodal journeys. Technical report, Karlsruhe Institute of Technology.

Dijkstra EW (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:2 69–271.

Donati AV, Montemanni R, Casagrande N, Rizzoli AE, Gambardella LM (2008). Time dependent vehicle routine problem with a multi ant colony system. European Journal of Operational Research 185(3), 1174-1191.

Fleming P, Corne D, Deb K, Knowles J (2003). The good of the many outweighs the good of the one: evolutionary multiobjective optimization. In coNNectionS 1(1), pages 9–13. IEEE Neur. Net. Soc.

Gaffney S and Smyth P (1999). Trajectory clustering with mixtures of regression models. In Proceedings of the 5th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 63–67.

Gallotti R and Barthelemy M (2015). The multilayer temporal network of public transport in Great Britain. *Scientific Data* **2**, Article number: 140056 doi:10.1038/sdata.2014.56

Geisberger R, Sanders P, Schultes D, Delling D (2008). Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In WEA, pages 319–333.

Gid'ofalvi G, Huang X and Pedersen T.B (2007). Privacy-preserving data mining on moving object trajectories. In Proceedings of the 8th IEEE International Conference on Mobile Data Management. IEEE, 60–68.

Greenfeld J.S (2002). Matching GPS observations to locations on a digital map. In Proceedings of the 81st Annual Meeting of the Transportaion Research Board, 576–582.

Gudmundsson J and Kreveld M.V. (2006). Computing longest duration flocks in trajectory data. In Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems. ACM, 35–42.

Gudmundsson J, Kreveld M.V, and Speckmann B (2004). Efficient detection of motion patterns in spatiotemporal data sets. In Proceedings of the 12th Annual ACM International Symposium on Advances in Geographic Information Systems. ACM, 250–257.

Hansen P (1979). Bicriteria path problems. In Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making - Theory and Application, 109–127.

Holme P and Saramäki J (2012). Temporal networks, *Phys. Rep.*, 519(3), 97–125.

Hart PE, Nilsson NJ, Raphael B (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Systems Science and Cybernetics, 4(2), 100–107.

Huang S.C, Jiau M.K, Lin C.H (2014). A Genetic-Algorithm-Based Approach to Solve Carpool Service Problems in Cloud Computing. IEEE Transactions on Intelligent Transpiration Systems, 16 (1), 352–364.

Huang S.C, Jiau M.K, Lin C.H (2014). Optimization of the Carpool Service Problem via a Fuzzy Controlled Genetic Algorithm. IEEE Transactions on Fuzzy Systems. 1(99), 1.

Jensen C.S., Lin D and Ooi B.C (2007). Continuous clustering of moving objects. IEEE Transaction on Knowledge and Data Engineering 19, 9 (2007), 1161–1174.

Jeung H, Shen H and Zhou X (2008). Convoy queries in spatio-temporal databases. In Proceedings of the 24th IEEE International Conference on Data Engineering. IEEE, 1457–1459.

Jeung H, Yiu M, Zhou X, Jensen C and Shen H (2008). Discovery of convoys in trajectory databases. Proceedings of the VLDB Endowment 1, 1 (2008), 1068–1080.

Jiang B, Yandong L, Ding Z, Eugene A, Hongyuan Z (2009). Learning to recognize reliable users and content in social media with coupled mutual reinforcement, Proceedings of the 18th international conference on World wide web, Madrid, Spain doi: 10.1145/1526709.1526717.

Jiau MK, Huang SC, Lin CH (2013). Optimizing the Carpool Service Problem with Genetic Algorithm in Service-Based Computing. In Services Computing (SCC), 2013 IEEE International Conference on (pp. 478–485). IEEE.

Jøsang, A.; Golbeck, J. (2009). Challenges for Robust of Trust and Reputation Systems. Proceedings of the 5th International Workshop on Security and Trust Management (STM 2009).. Saint Malo, France.

Jøsang A, Ismail R, BoydA C (2007). Survey of trust and reputation systems for online service provision *Decision Support Systems*, 43 (2), 618–644.

Jurca R, Faltings B (2003). An incentive compatible reputation mechanism. In: *Proceedings of the IEEE International Conference on E-Commerce* (CEC), 285–292.

Lazzari, M. (2010). An experiment on the weakness of reputation algorithms used in professional social networks: the case of Naymz. Proceedings of the IADIS International Conference e-Society.. Porto, Portugal.

Lee, J. G., J. Han, and K. Y. Whang. (2007). Trajectory clustering: A partition-and-group framework. In Proceedings of the ACM SIGMOD Conference on Management of Data. ACM, 593–604.

Lee, J., J. Han, and X. Li. (2008). Trajectory outlier detection: A partition-and-detect framework. In Proceedings of the 24th IEEE International Conference on Data Engineering. IEEE, 140–149.

Lee W.-C. and J. Krumm. (2011). Trajectory preprocessing. Computing with Spatial Trajectories, Y. Zheng and X. Zhou (Eds.). Springer, 1–31.

Lee, Jae-Gil, Jiawei Han (2007). Trajectory Clustering: A Partition-and-Group Framework. SIGMOD07 Proceedings of the 2007 ACM SIGMOD international conference on Management of data, 593-604.

Li, Z., B. Ding, J. Han, and R. Kays. (2010). Swarm: Mining relaxed temporal moving object clusters. Proceedings of the VLDB Endowment 3, 1–2 (2010), 723–734.

Liao H, Zeng A, Xiao R, Ren Z-M, Chen D-B, et al. (2014) Ranking Reputation and Quality in Online Rating Systems. *PLoS ONE*, 9(5): e97146. doi: 10.1371/journal.pone.0097146

Liu L (2010). Data models and algorithms for multimodal route planning with transportation networks. PhD thesis, Technische Universitaet Muenchen.

Möhring RH, Schilling H, Schütz B, Wagner D, Willhalm T (2005). Partitioning graphs to speed up dijkstra's algorithm. In WEA, 189–202.

Montemanni R, Gambardella LM (2004). An exact algorithm for the robust shortest path problem with interval data. Computers and Operations Research 31(10), 1667-1680.

Montemanni R, Gambardella LM (2005). The robust shortest path problem with interval data via Benders decomposition. 4OR 3(4), 315-328.

Montemanni R, Barta J, Mastrolilli M, Gambardella LM (2007). The robust traveling salesman problem with interval data. Transportation Science 41(3), 366-381.

Morzy, Mikołaj (2008). New algorithms for mining the reputation of participants of online auctions.Algorithmica, 52.1: 95-112.

Nergiz, M. E., M. Atzori, Y. Saygin, and B. Guc. (2009). Towards trajectory anonymization: A generalization based approach. Transactions on Data Privacy 2, 1 (2009), 47–75.

Newson,P., J. Krumm. (2009). Hidden Markov map matching through noise and sparseness. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Geographical Information Systems. ACM, 336–343.

Ochieng, W. Y. , M. A. Quddus, and R. B. Noland. (2004). Map-matching in complex urban road networks. Brazilian Journal of Cartography 55, 2 (2004), 1–18.

Pajor T (2009). Multi-modal route planning. Diploma thesis, Karlsruhe Institute of Technology.

Paniza-Fullana A. (2010). Security Measures in the Personal Data Protection Rules: Technological Solutions and Legal Adaptation, Advanced Research on Information Security and Privacy: Deliverable.

Pink, O. and B. Hummel. (2008). A statistical approach to map matching using road network geometry,topology and vehicular motion constraints. In Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems. IEEE, 862–867.

Poole RW, Balaker T (2005). Virtual exclusive busways: improving urban transit while relieving congestion Los Angeles, CA: Reason Foundation.

Pyrga E, Schulz F, Wagner D, Zaroliagis CD (2007). Efficient models for timetable information in public transportation systems. ACM Journal of Experimental Algorithmics, 12.

Rainie, L. and Wellman, B., (2012). Networked: The New Social Operating System. MIT Press.

Resnick P, Zeckhauser R (2002). Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. In: Baye, M.R. (Ed.). The Economics of the Internet and E-Commerce, 11. 127–157.

Small KA, Winston C, Yan J, Baum-Snow N, Gómez-Ibáñez JA (2006). Differentiated Road Pricing, Express Lanes, and Carpools: Exploiting Heterogeneous Preferences in Policy Design [with Comments]. Brookings-Wharton Papers on Urban Affairs, 53–96.

Slee, T. (2013). Some Obvious Things About Internet Reputation Systems.

Song S, Hwang K, Zhou R, Kwok Y-K (2005). Trusted P2P Transactions with Fuzzy Reputation Aggregation, *IEEE Internet Computing*, 9(6), 24-34.

Strasser B (2012). Delay-robust stochastic routing in timetable networks. Master's thesis, Karlsruhe Institute of Technology.

Su, H., K. Zheng, H.Wang, J. Huang, and X. Zhou. (2013). Calibrating trajectory data for similarity-based analysis. In Proceedings of the 39th International Conference on Very Large Data Bases. VLDB Endowment, 833–844.

Tang, L. A., Y. Zheng, J. Yuan, J. Han, A. Leung, C. Hung, and W. Peng. (2012). Discovery of traveling companions from streaming trajectories. In Proceedings of the 28th IEEE International Conference on Data Engineering. IEEE, 186–197.

Tang, L. A., Y. Zheng, J. Yuan, J. Han, A. Leung, W. Peng, and T. L. Porta. (2012). A framework of traveling companion discovery on trajectory data streams. ACM Transactions on Intelligent Systems and Technology 5, 1 (2012).

Terrovitis, M. and N. Mamoulis. (2008). Privacy preservation in the publication of trajectories. In Proceedings the 9th IEEE International Conference on Mobile Data Management. IEEE, 65–72.

Timothy, S., A. Varshavsky, A. Lamarca, M. Y. Chen, and T. Chounhury. (2006). Mobility detection using everyday GSM traces. In Proceedings of the 8th International Conference on Ubiquitous Computing. ACM, 212–224.

Xia, J., Curtin, K.M., Li, W., Zhao, Y. (2015). A new model for a carpool matching service. PLoS ONE 10(6).

Xue, A. Y., R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu. (2013). Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In Proceedings of the 29th IEEE International Conference on Data Engineering. IEEE, 254–265.

Ye, Y., Y. Zheng, Y. Chen, J. Feng, and X. Xie. 2009. Mining individual life pattern based on location history. In Proceedings of the 10th IEEE International Conference on Mobile Data Management. IEEE, 1–10.

York B,.Fabricatore D. (2001) Puget Sound Vanpool Market Assessment. Washington State Dept Transport

Yu, H., Lu F (2012). Advanced multi-modal routing approach for pedestrians. In Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, 2349 –2352.

Yuan, J., Y. Zheng, L. Zhang, X. Xie, and G. Sun. (2011). Where to find my next passenger? In Proceedings of the 13th International Conference on Ubiquitous Computing. ACM, 109–118.

Yuan, N. J., Y. Zheng, L. Zhang, and X. Xie. (2013). T-Finder: A recommender system for finding passengers vacant taxis. IEEE Transaction on Knowledge and Data Engineering 25, 10 (2013), 2390–2403.

Yuan, J., Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. (2010). T-Drive: Driving directions based on taxi trajectories. In Proceedings of the 18th Annual ACM International Conference on Advances in Geographic Information Systems. ACM, 99–108.

Yuan, J., Y. Zheng, X. Xie, and G. Sun. (2011). Driving with knowledge from the physical world. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 316–324.

Yuan, J., Y. Zheng, X. Xie, and G. Sun. (2013). T-Drive: Enhancing driving directions with taxi drivers' intelligence. IEEE Transaction on Knowledge and Data Engineering 25, 1 (2013), 220–232.

Yuan, N. J. Y. Zheng, L. Zhang, and X. Xie. (2013). T-Finder: A recommender system for finding passengers and vacant taxis. IEEE Transaction on Knowledge and Data Engineering 25, 10 (2013), 2390–2403.

Winters P, Axton R, Gunnell J (1991). Transit and Ridesharing Information Study. *Transportation Research Record*, 1321Washingon D.C.

Witkowski J, Seuken S, Parkes D.C (2011). Incentive-compatible escrow mechanisms. In: *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI), 751–757.

Wu Q, Zhang X, Zhang M, Lou Y, Zheng R, and Wei W (2014). Reputation Revision Method for Selecting Cloud Services Based on Prior Knowledge and a Market Mechanism. *The Scientific World Journal*, doi: 10.1155/2014/617087, 9.

Zacharia, Maes (2000). Trust management through reputation mechanisms, *Applied Artificial Intelligence: An International Journal*, 14(9), 881-907.

Zheng Yu (2015). Trajectory data mining: An overview. ACM Trans. Intell. Syst. Technol. 6, 3, Article 29 (May 2015), 41 pages. DOI: http://dx.doi.org/10.1145/2743025.

Zheng, Y., Y. Chen, X. Xie, and W.-Y. Ma. (2009). GeoLife2.0: A location-based social networking service. In Proceedings of the 10th IEEE International Conference on Mobile Data Management. IEEE, 357–358.

Zheng, Y. and X. Xie. (2011). Learning travel recommendations from user-generated GPS traces. ACM Transactions on Intelligent Systems and Technology 2, 1 (2011), 2–19.

Zheng, Y., Q. Li, Y. Chen, and X. Xie. (2008). Understanding mobility based on GPS data. In Proceedings of the 11th International Conference on Ubiquitous Computing. ACM, 312–321.

Zheng, Y., L. Liu, L.Wang, and X. Xie. (2008). Learning transportation mode from raw GPS data for geographic application on the Web. In Proceedings of the 17th International Conference on World Wide Web. ACM, 247–256.

Zheng Z, Lyu MR (2008). Ws-Dream: A Distributed Reliability Assessment Mechanism for Web Services, *Proc. 38th Int. Conf Dependable Systems and Networks (DSN , '08)*, 392-397.

Zhu, Y. , Y. Zheng, L. Zhang,D. Santani, X. Xie, andQ. Yang. (2011). Inferring Taxi Status using GPS Trajectories. Technical Report MSR-TR-2011-144.